

Máster en Programación avanzada en Python para Big Data, Hacking y Machine Learning

Programación Python para Machine Learning

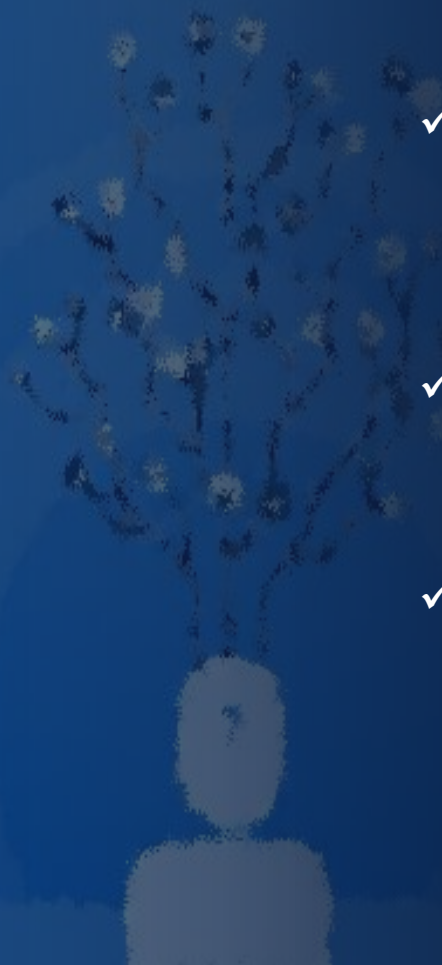
LECCIÓN 09

Lección 9: Árboles de Decisión.

ÍNDICE

- ✓ Introducción
- ✓ Objetivos
- ✓ Principios teóricos y conceptos de los Árboles de decisión.
- ✓ Implementación de un Árbol de decisión.
- ✓ Consideraciones a tener en cuenta.
- ✓ Conclusiones

INTRODUCCIÓN



- ✓ Modelo intuitivo, simple, no lineal, interpretable y poderoso.
- ✓ Regresión y clasificación.
- ✓ Árboles de decisión: modelo de Machine Learning supervisado no lineal.

OBJETIVOS

Al finalizar esta lección serás capaz de:

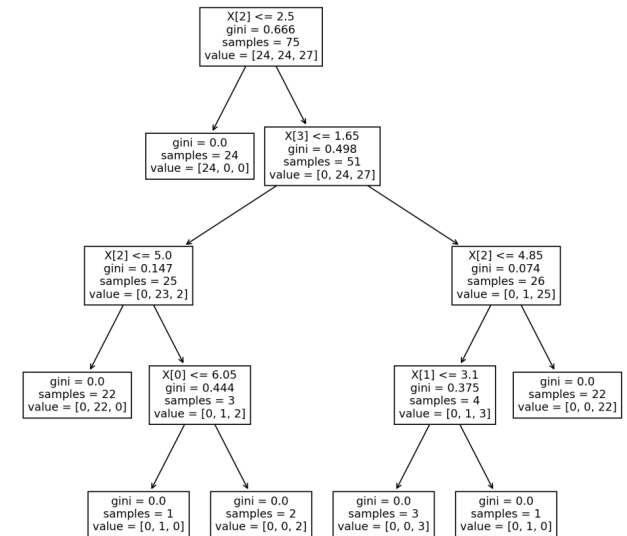
- 1 Conocer los principios en los que se basan las Redes Neuronales Artificiales.
- 2 Reconocer los componentes de un Árbol de decisión y entender cómo se construye.
- 3 Implementar un modelo de Árbol de decisión en Python para resolver problemas de clasificación y regresión.
- 4 Identificar los aspectos a tener en cuenta para mejorar el rendimiento de un Árbol de decisión.

ÁRBOLES DE DECISIÓN

Modelo supervisado no lineal para regresión y clasificación.

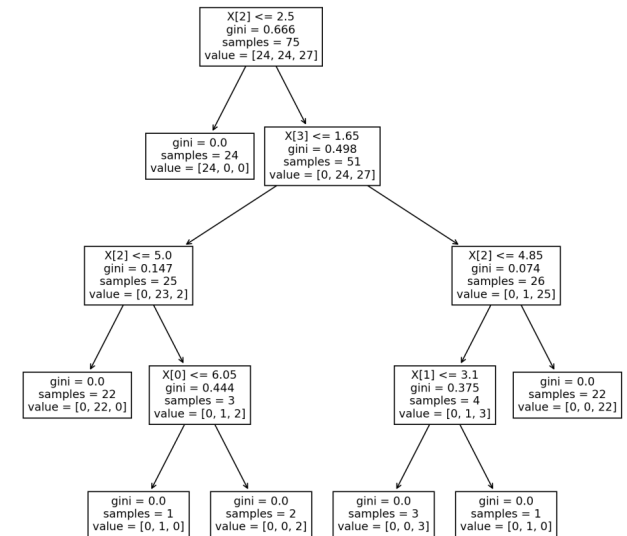
Ir generando particiones binarias de los datos de

Cada nueva partición genera un subgrupo de datos lo más homogéneo posible.



ÁRBOLES DE DECISIÓN

- ✓ Los nodos internos representan cada una de las características a considerar para tomar una decisión y generar la partición.
- ✓ Las ramas representan la decisión en función de la condición del nodo del que parten.
- ✓ Los nodos hoja representan el resultado final de la decisión, es decir, la predicción.
- ✓ Nodo raíz y profundidad.

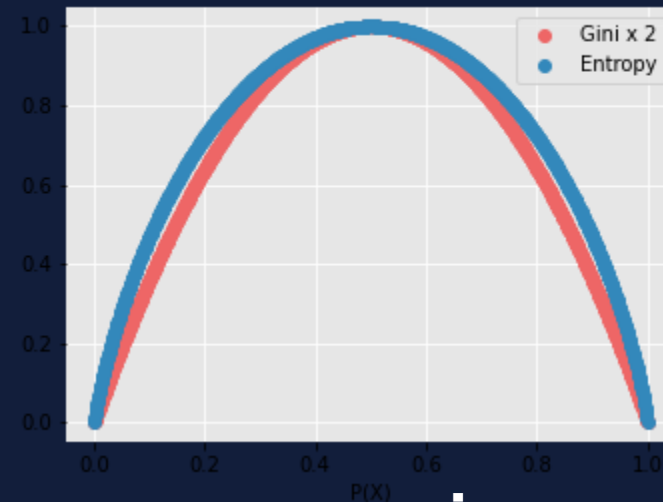
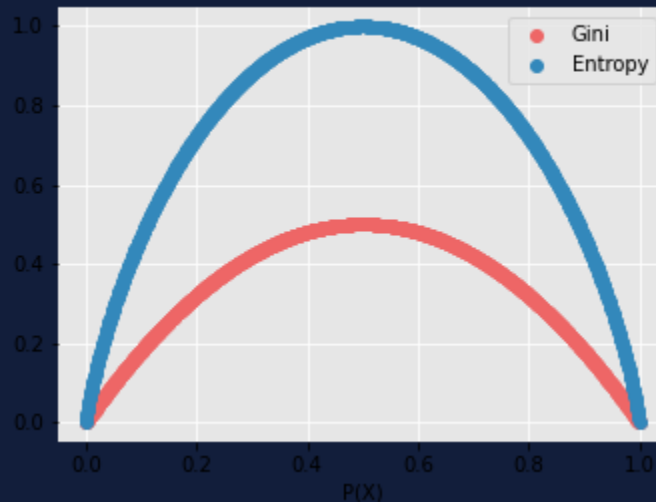


ÁRBOLES DE DECISIÓN

Cuantificar la homogeneidad :

Clasificación	Índice de Gini
	Entropía (Ganancia de Información)
Regresión	RMSE
	Error absoluto medio

Función de costo: al nodo padre se le asigna un promedio ponderado de la métrica de homogeneidad de sus dos nodos hijo.



ÁRBOLES DE DECISIÓN

Entrenamiento de un árbol de decisión:

1. Primera partición (nodo raíz) se toman todas las características del conjunto de entrenamiento y, para cada una de ellas, se definen todos los posibles umbrales. Se considera umbral cada punto intermedio entre dos valores consecutivos de cada característica.
2. Para cada umbral, se calcula la partición y la métrica de homogeneidad de cada hijo. Con estos valores, la función de costo del padre.
3. Se elige el umbral con menor función de costo.
4. El proceso se repite de modo iterativo hasta conseguir todos los nodos hoja.

✓ Algoritmo voraz

ÁRBOLES DE DECISIÓN

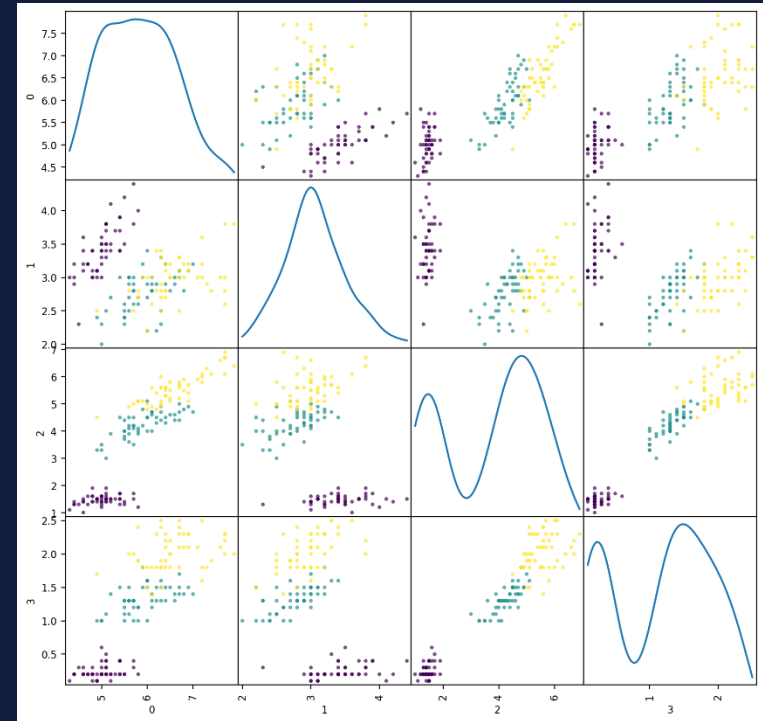
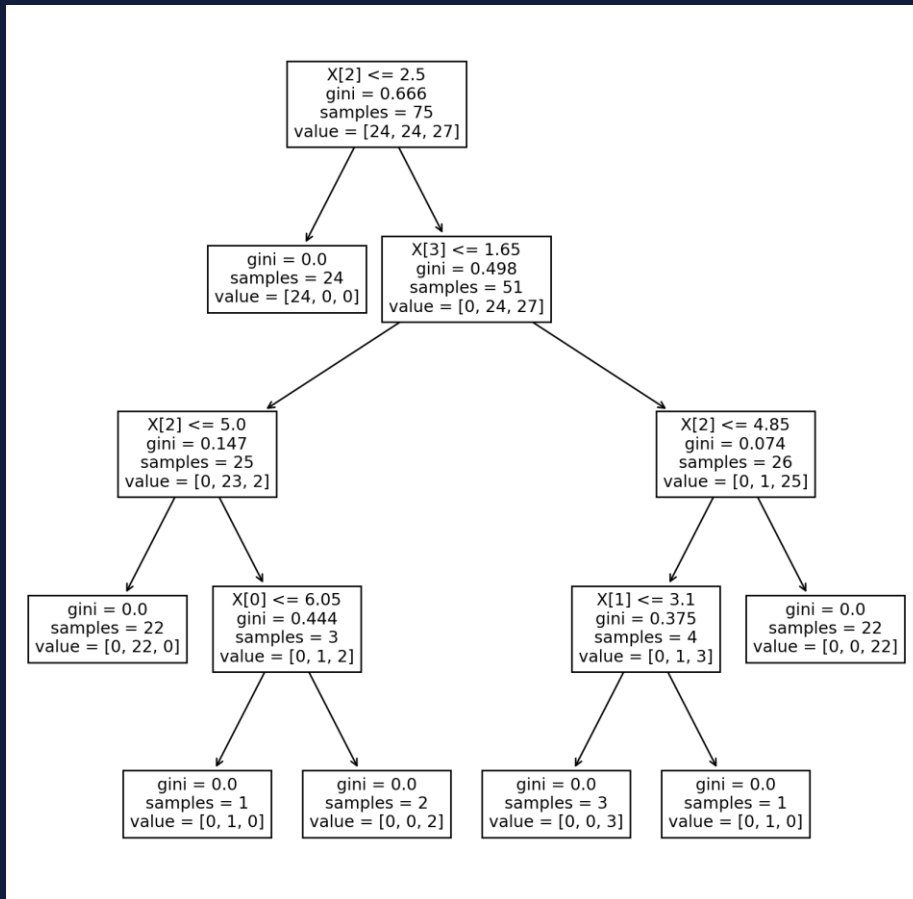
Árboles que todos los nodos hoja
“puros” → overfitting.

Dos posibles estrategias para evitarlo:

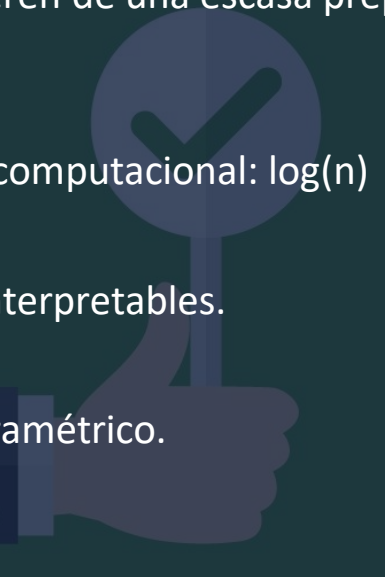
- ✓ Estrategia pre-pruning.
- ✓ Estrategia post-pruning.

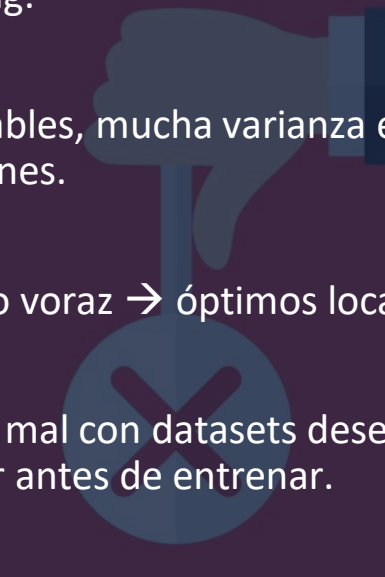


ÁRBOLES DE DECISIÓN: Interpretabilidad



ÁRBOLES DE DECISIÓN

- ✓ Requieren de una escasa preparación de los datos.
 - ✓ Coste computacional: $\log(n)$
 - ✓ Muy interpretables.
 - ✓ No paramétrico.
- 

- ✓ Overfitting.
 - ✓ Poco estables, mucha varianza en sus predicciones.
 - ✓ Algoritmo voraz \rightarrow óptimos locales.
 - ✓ Funciona mal con datasets desequilibrados. Equilibrar antes de entrenar.
- 

MUCHAS GRACIAS POR SU ATENCIÓN



jperez@grupomainjobs.com



Javier Pérez Rodríguez
www.linkedin.com/in/perezxavi



twitter.com/eiposgrados



facebook.com/eiposgrados



instagram.com/eiposgrados